DOI 10.5162/sensor2015/P9.2

Fast and Efficient Dual-Forecasting Algorithm for Wireless Sensor Networks

Florian Strakosch[1], *Faouzi Derbel*[1]
[1] *Leipzig University of Applied Sciences, Wächterstraße 13, 04107 Leipzig, Germany,*
Florian.Strakosch@HTWK-Leipzig.de

Abstract:

Data transmission is by far the most energy consuming operation a wireless sensor node needs to perform. Hence, the most effective method of extending the lifetime of such a usually battery-powered device is to reduce the amount of sending intervals. One way of treating this issue is by using system identification and forecasting. That basically implies to take a reasonable amount of sample data for building a close-to-perfect system model and to perform a one-step prediction out of the given information on sensor node and gateway simultaneously. For doing so measurements of temperature, humidity, force and friction have been taken and analyzed. The novelty of our approach is the simplicity of our algorithm compared to other works, its measureable energy savings in real life environments and its potential for online parameter estimation.

Key words: system identification, parallel forecasting, one-step prediction, lifetime extension, wireless sensor networks.

Introduction

Nowadays, the amount of possible fields of application for wireless sensor networks (WSNs) is increasing ever faster with seemingly endless ideas. While the principle concept was just measuring environmental values in the beginning, complex surveillance and automation tasks came into consideration more often recently. Thinking of industrial plants for example, to determine vibrations, forces, friction and other parameters of conveyer belts can be very tough especially when trying to find a wired solution. Hence, the availability of wireless sensors is a welcoming change in this particular area of application.

Although this advanced technology paves the way for a wide variety of possibilities, it comes with some new challenges as well. Speaking of which their limited resources in terms of size, availability, computation time and storage capacity are restrictions that need to be obeyed. One of the most demanding is the available energy, though. In most cases WSNs are battery powered and therefore have a limited lifetime. Its span mainly depends on the task the sensor has to fulfill and the capacity of the power sources, and can reach from several weeks up to ten or more years.

For making WSNs even cheaper, long-lasting and therefore more attractive to customers the efficient usage of the battery for either using smaller cells or extending the lifetime is one major focus in research at the moment. Since the most energy consuming component of a sensor node is the transceiver as shown in Fig. 1, reducing the amount of transmissions over time is a common approach.

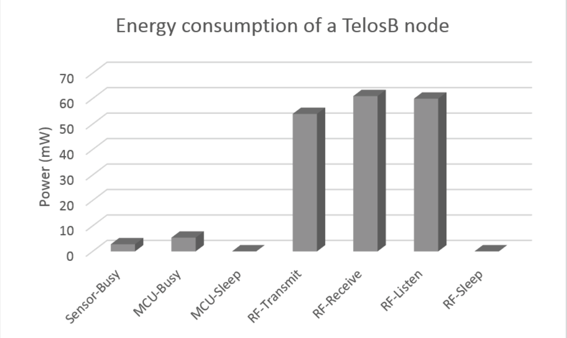

Fig. 1. Energy consumption of wireless sensor nodes' components.

Problem

Several well-known data aggregation strategies have been implemented to aim for a reduced amount of current drawn from the battery in the past, but major improvements just started with the idea of using system identification and forecasting [1, 2] in 2001. The concept was to describe an observed physical process with the help of mathematics, combine this information with the sensors inputs to predict the next value and to not transmit successful forecasted measurement.

Transferring the load from the transceiver to the microcontroller made sense especially when looking at Fig. 1 again, but there were a few drawbacks to this method. First of all the knowledge of the physics of the whole process or a large amount of sample data for building an adequate model is required. Once the model is build it might became too complex for being implemented on a relatively cheap sensor node. Further works [3, 4, 5, 6] tried to address this and similar issues but either designed solutions for very specific purposes or remained in conceptual state.

Hence, the focus of our work was to find a simple, fast and efficient solution for implementing a forecasting algorithm into sensor nodes measuring a large variety of different values in factory sites without having extensive a-priori knowledge. In addition, results should be verified in a real life environment using TelosB sensor nodes.

Solution

The principle idea of or work is based on the approaches mentioned above and the routine is shown in Fig. 2. The concept starts with having the ability of almost simultaneously measuring and calculating. This being said, the nodes task is to sense for new input while performing a one-step prediction out of a limited amount of previous data at the same time. Afterwards, the measured and the forecasted value are compared against a tight threshold. In case of the calculation is close enough the result is being used for the next prediction and the sensed value is dropped. Otherwise the measurement needs to be transmitted and the prediction result will be ignored.

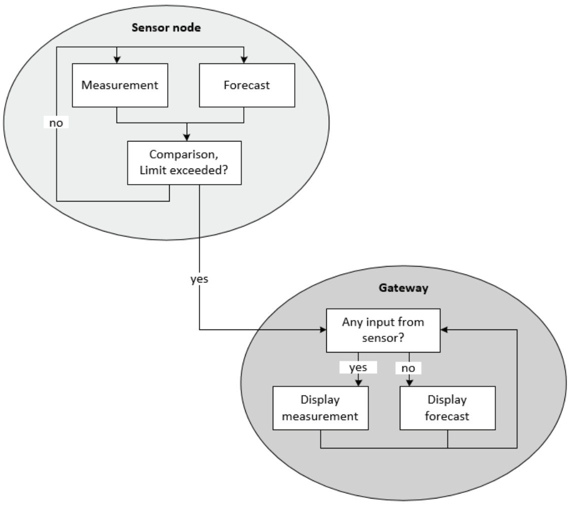

Fig. 2: Fast and efficient dual-forecasting algorithm scheme

Meanwhile the gateway is performing the same (dual) forecasting algorithm as long as there is no contrary input from the sensor node recognized.

In terms of the system model describing the process a linear 5th order AutoRegressive model with eXogenous input (ARX) was chosen. The reasons for that were mainly its simplicity concerning building as well as implementing and its exactable performance even in difficult fields according to [7]. For the initial determination of the corresponding parameters a set of sample data in form of temperature readings was taken.

Basics

ARX in general is a linear difference equation that describes the connectedness between the output $y(t)$ and the input $u(t)$ as follows:

$$\begin{aligned} &y(t)+a_1\cdot y(t-1)+...+a_n\cdot y(t-n)\\ &=b_1\cdot u(t-1)+...+b_m\cdot u(t-m) \end{aligned} \tag{1}$$

Since the measurements are collected by sampling, discrete time t can be assumed. To keep the required system knowledge to a minimum the influence of noise is neglected, what in turn helps to keep the later algorithm simple as well as fast.

In order to perform a one-step prediction, which means calculating a new output value out of previous data, (1) can be rewritten into:

$$\begin{aligned} &y(t)=-a_1\cdot y(t-1)-...-a_n\cdot y(t-n)\\ &+b_1\cdot u(t-1)+...+b_m\cdot u(t-m) \end{aligned} \tag{2}$$

Imagining a real life environment one will soon realize, that we only have access to the output value and the sampling rate. Combining this fact with the statement of neglecting noise influences $u(t)$ can be assumed to equal $y(t)$, which leads us to:

$$\begin{aligned} &y(t)=\\ &-a_1\cdot b_1\cdot y(t-1)-...-a_n\cdot b_m\cdot y(t-n) \end{aligned} \tag{3}$$

Now we do not need two different parameters anymore. Substituting $a_n\cdot b_m$ by a_n reduces (3) to:

$$y(t)=-a_1\cdot y(t-1)-...-a_n\cdot y(t-n) \tag{4}$$

This equation is already good enough for implementing it into a sensor node but not as suitable for system identification purposes. Hence, the vectors $\vec{\theta}$ and $\vec{\phi}$

$$\vec{\theta} = [-a_1 \ldots -a_n]^T \qquad (5)$$
$$\vec{\phi}(t) = [y(t-1) \ldots y(t-n)]^T$$

are introduced to note (4) in a more compact way as:

$$y(t) = \vec{\phi}^T(t) \cdot \vec{\theta} \qquad (6)$$

Thereby $\vec{\theta}$ contains the ARX parameters (weights) that need to be determined next. For doing so the mentioned set of sample data collected within the time interval $1 \le t \le N$ is used in combination with the least square method according to [6] and [8] in from of:

$$\vec{\theta}_N = (\sum_{t=1}^{N} \vec{\phi}(t) \cdot \vec{\phi}^T(t))^{-1} \cdot \sum_{t=1}^{N} \vec{\phi}(t) \cdot y(t) \qquad (7)$$

Energy Savings

As described above the goal of system identification and forecasting is to extend the lifetime of WSNs by reducing the current drawn from the battery over time. Hence, an equation for describing the possible energy saving is needed. In order to get easily comparable results we assume that the gateway is always ready for reception and therefore concentrate on the sender only according to [9].

Consequently, the following representation for energy consumption is used:

$$E = U \cdot I_0 \cdot t_0 + U \cdot I_{tx} \cdot (t_{oh} + p_{data} \cdot t_{byte}) + (r-1) \cdot U \cdot I_{CPU} \cdot t_{pred} \qquad (8)$$

In this case U is the voltage of the battery, I_0 and I_{tx} are the currents drawn while the microcontroller is sleeping and during transmission, while t_0 represents the sleeping time. Accordingly, t_{oh} and t_{byte} are the times needed for transmitting the overhead and one byte of data, p_{data} is the amount of bytes, I_{CPU} and t_{pred} are current and time during the prediction calculation and lastly r is the number of the forecasting steps.

In case the algorithm helps to avoid $r-1$ out of r transmission the normalized energy saving function can be noted as:

$$\Delta E(r) = \frac{r \cdot E_{std-trans} - E_{pred-trans}}{r \cdot E_{std-trans}} \qquad (9)$$

Here $E_{std-trans}$ is the energy required for a standard transmission and $E_{pred-trans}$ for performing a one-step prediction respectively. Substituting (8) into (9) one will get:

$$\Delta E(r) = 1 - \left(\frac{I_0 \cdot t_0 + I_{tx} \cdot (t_{oh} + p_{data} \cdot t_{byte})}{r \cdot (I_0 \cdot t_0 + I_{tx} \cdot (t_{oh} + p_{data} \cdot t_{byte}))} + \frac{(r-1) \cdot I_{CPU} \cdot t_{pred}}{r \cdot (I_0 \cdot t_0 + I_{tx} \cdot (t_{oh} + p_{data} \cdot t_{byte}))}\right) \qquad (10)$$

$\Delta E(r)$ needs to be > 0 to proof that energy is saved, which can only be realized by fulfilling the condition $r > 1$.

Using the technical data $I_0 \approx I_{tx} \approx$ 20mA, $t_0 \approx$ 0.4ms, $t_{byte} \approx$ 0.032ms, $t_{oh} \approx$ 0.8ms, $I_{CPU} \approx$ 2mA and $t_{pred} \approx$ 1ms of the mentioned TelosB sensor node by Crossbow including the assumption that the overhead is 25 Bytes, a measurement 2 Bytes and its timestamp 4 Bytes in length as well as $p_{data} =$ 6, (10) will become:

$$\Delta E(r) = \frac{(r-1) \cdot 1.292}{r \cdot 1.392} \qquad (11)$$

This leaves r the only unknown parameter and therefore the potential energy saving can be determined directly out of the amount of successful forecasted values.

Results

Using (7), a set of 3000 temperature readings covering three days, 1000 validation measurements and the MATLAB System Identification Toolbox we were able to find some well-fitting parameters for the intended 5th order ARX model as follows:

$$y(t) = 1.2669 \cdot y(t-1) - 0.4078 \cdot y(t-2) + 0.1899 \cdot y(t-3) - 0.0310 \cdot y(t-4) - 0.0181 \cdot y(t-5) \qquad (12)$$

To verify the quality of this weights another 6000 sample data were collected and used as a reference. As one can see in Fig. 3 the predicted values are “following” the measured ones quite well, leading to a prediction probability of more than 80%. Hence, in theory 4 out of 5 transmissions can be avoided. In other words: $r = 5$.

This in turn means that an energy saving potential of remarkable 74% can be reached according to (11).

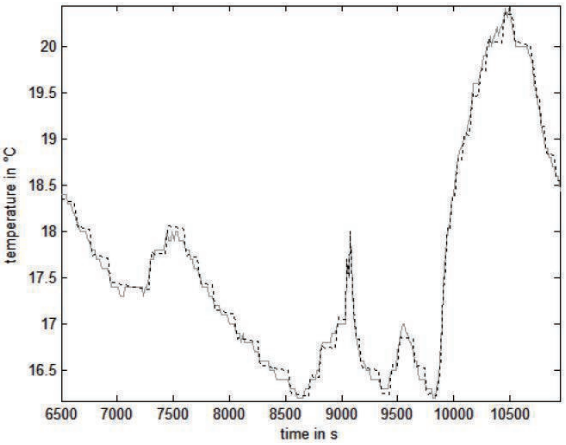

Fig. 3: Comparison between measured (grey) and forecasted (black) values

Still, building a weighted sum out of 3000 temperature data with the help of numerical software is not necessary a proof of concept. That is why we went on analyzing three major improvements to our work so far: a) using (12) for other measurements, b) reducing the amount of sample data and repeating the system identification as well as verification and finally c) extending the given variety of measurements of conveyer belts. More complex models like the Kalman can be neglected here, because their forecasting accuracy is not higher for linear one-step prediction in this particular case [10].

As for a) we figured that (12) can easily be used for temperature readings with a wider range as well as for similar (corresponding) measurements like humidity and dew point. In both cases the forecasting probability and therefore the energy saving differs in the range of 3% maximum compared to a specific system identification. This finding can save engineering time and increases usability and applicability of WSNs. On the other hand (12) will not work for force and friction in the same manner but will cause only half of the energy saving possible.

In terms of b) using only one third of the original samples can still work out with up to 70% of energy saving potential. Reducing the amount to 100 readings will result in only 20 – 40 % of energy saving. Therefore an online system identification process might become possible this way.

Working on c) it was found that friction can also be handled quit well by a 5th order ARX model. Since the data coming from the sensor is very noisy the result is not as good as for temperature readings, though. The energy saving potential reaches up to 60% adding a moving average as signal preprocessing unit to the algorithm.

The findings of a) have already been verified with less than 5% difference in a real life environment using the described TelosB sensor nodes.

Conclusion / Prospect

As of now, we were able to successfully forecast temperature, humidity, dew point, friction and elongation measurements of a carrier belt system. This was achieved by the following two steps: a) taking sample data sets from the sensors and building a 5th order ARX-model using MATLAB, b) performing the same (dual) prediction algorithm on sensor and gateway as shown in Fig. 2. Therefore, the wireless node can compare the measurement and the forecast against a tight threshold and only transmits its data when the given limit is exceeded.

Since an ARX-model is basically a weighted sum, it is a fairly simple and efficient algorithm to build and it does hardly require any computation time or energy. Still, it is a powerful tool with the ability to outperform other models like ARIMA and the Kalman-Filter in some cases. Hence, we can extend the nodes' lifetime by up to 74% for temperature and 60% for friction, using TelosBs and TinyOS.

In addition, we are working on reducing the sample data length for performing step a) online directly on the sensor nodes in an efficient way with partial only slightly worse results.

Acknowledgement

The authors would like to thank the ESF for founding this work as part of the junior research group "AIS – Autarkic Intelligent Sensor Networks".

References

[1] S. Goel, T. Imielinski, *ACM SIGCOMM Computer Communication Review - Special issue on wireless extensions to the internet, Volume 31, Issue 5,* 82-98 (2001); doi: 10.1145/1037107.1037117

[2] F. Derbel, Modeling fire detector signals by means of system identification techniques, *IEEE Transactions on Instrumentation and Measurement*, vol. 50 (2001)

[3] E.-O. Blaß, J. Horneber, M. Zitterbart, *Vehicular Technology Conference Spring 2008*, 86-87 (2008); doi: 10.1109/VETECS.2008.30

[4] Y.-Aë. Le Borgne, et al., Adaptive model selection for time series prediction in wireless

sensor networks, *Signal Process* (2007); doi: 10.1016/j.sigpro.207.05.015

[5] G. Wei, Y. Ling, B. Guo, B. Xiao, A. V. Vasilakos, *Computer Communications* 34, 793-802 (2011); doi: 10.1016/j.comcom.2010.10.003

[6] F. Derbel, *9th International Multi-Conference on Systems, Signals and Devices (SSD)*, 1-6 (2012); doi: 10.1109/SSD.2012.6198105

[7] J. Zhao, H. Liu, Z. Li, W. Li, *Advances in Wireless Sensor Networks*, 695-701 (2012); doi: 10.1007/978-3-642-36252-1_65

[8] L. Ljung, *System Identification - Theory for the user*, 2nd ed. (1999)

[9] F. Derbel, Smart wireless sub-metering (2010)

[10] F. Strakosch, F. Derbel, Energy-Saving Forecasting Techniques for Measurement Data Transmitting WSNs, *5th IMEKO TC19 SYMPOSIUM,* 133-137 (2014)